UNIVERSIDAD POLITÉCNICA DE MADRID
E.T.S. de Ingenieros Aeronáuticos

POLITÉCNICA

Accelerating numerics on PDEs using POD and Galerkin projection

Filippo Terragni, María-Luisa Rapún, and José M. Vega

DSPDES'10
Barcelona, June 3,2010

Main goal: to accelerate numerical codes.

- There is a current trend to replace wind tunnel tests by CFD in Industrial Aerodynamics.
- But computational fluid dynamics of 3D problems at large Re may require non affordable resources. Same trend in other fields.
- Example: Commercial aircraft, L=80-meter-long span, cruising at v=250 meters per second, at an altitude of 10,000 meters. The Reynolds # is ~10^9. Direct numerical simulation requires more than $Re^{9/4} \sim 10^{20}$ grid points and a time step smaller than $L/(Re^{1/2}v) \sim 10^{-5}$ seconds. Current algorithms and software, with a supercomputer (10^{14} floating-point operations per second) would take several thousand years to compute the flow for just one second of flight time!
- Cheaper alternatives: RANS approximates cruise conditions as steady states (2 CPU days in a PC). Panel (1 CPU minute) and AVL (1 CPU second) methods are quite rough.

Question: Are there better (much cheaper, yet sufficiently precise) alternatives?

Answer : Better algorithms, better software/hardware, reduced order models (ROMs).

Main goal: to accelerate numerical codes (cont'ed).

- Additional difficulty: Design and certification (as other industrial/scientific problems do) involve many parameters, say various tens. Multi-parameter settings exponentially enlarge computational effort.
- Good news: The required precision is usually not quite large in industrial problems.
- Industrial solvers are usually fairly rough (coarse meshes, unphysical terms/BCs added to accelerate/stabilize), but still require a large number of mesh points/modes/time steps.
- Such numerical complexity (# of degrees of freedom) is much larger than the physical complexity of the flow (# of qualitatively different spatio-temporal features): a description in terms of a few modes should be possible.

Question: Is there a good set of modes for each specific problem?

Answer: Numerically calculated snapshots + POD.

Caution: Precision of the numerical solver (quality of the snapshots) matters!

Summary of the talk:

- Already known: POD+Galerkin projection (attractors).
- New: Local POD+Galerkin projection (transients+attractors).
- Application to the complex Ginzburg-Landau eq’n.
- Application to the pulsating cavity.
- Some conclusions.

POD+Galerkin projection

- The proper orthogonal decomposition (POD) of a system of vectors provides an (RMS) optimal basis of the vector span.
- If the vector system shows redundancies (due to, e.g., physical laws), then the dimension of the truncated (for a given precision) POD basis is much smaller than that of the vector system.

Main idea in two steps:

1. Numerical calculation of some representative snapshots (Sirovich, 1978) in a given time interval.
2. Galerkin projection of the system of PDEs on the n most energetic POD modes, generated from the set of snapshots. Due to spatio-temporal redundancies, n is usually “small”.

The resulting system of ODEs is called reduced order model (ROM).

POD+Galerkin projection (cont'ed)

Advantages:

- Reduces the dimension of the problem.
- Can be used (in principle) for parameter values not considered in the snapshot set calculation.

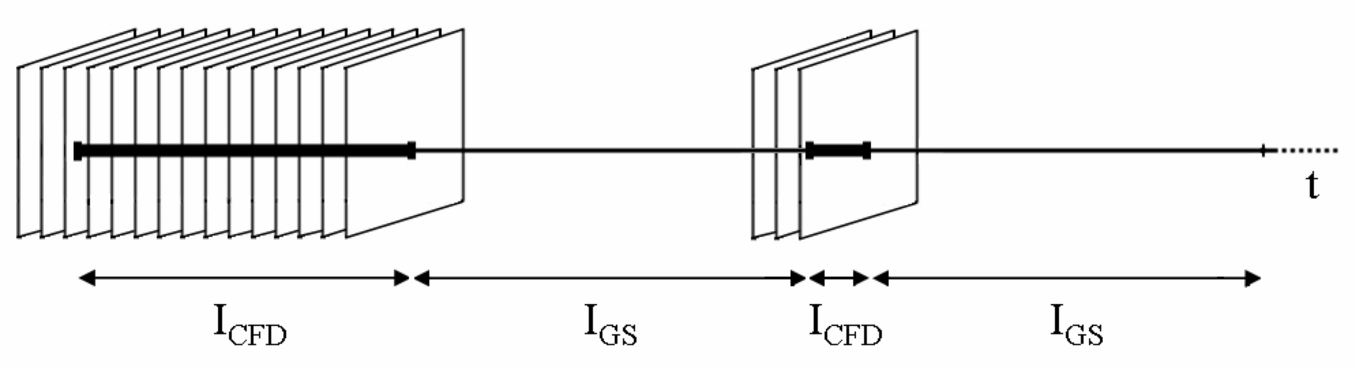
Difficulties:

- Each (L_2) projection requires all mesh points (expensive). Overcome when nonlinearity is algebraic.
- Non-homogenous BCs accounted for through a change of variable, which may require some re-meshing when staggered grids are used.
- Not suitable for transients.
- Shows instability due to higher order modes truncation.

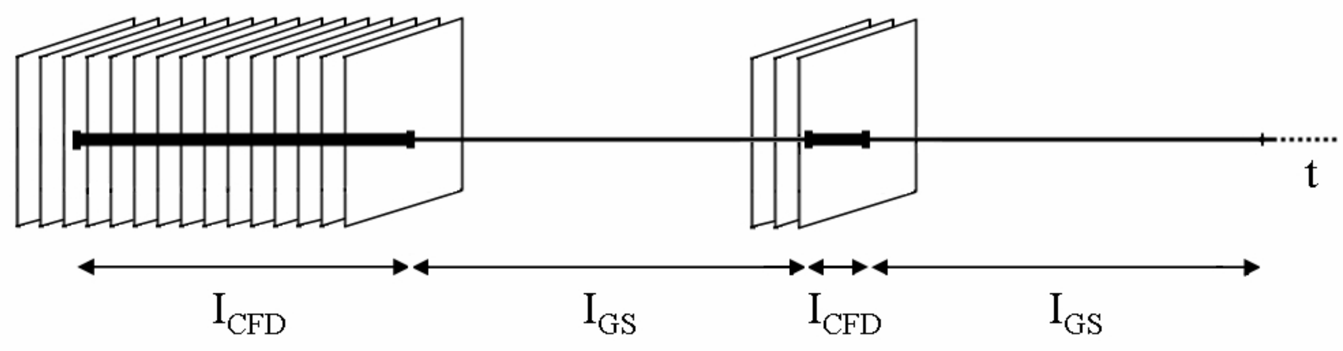
Local POD+Galerkin projection.

Rapun & V., J. Comp. Phys., 229(2010), 3046-3063.

- Originally developed to accelerate time dependent solvers in 1D.
- Can be extended to treat steady flows (RANS solver) for varying parameter values.
- Combines the CFD solver with a Galerkin system in interspersed time intervals, I_{CFD} and I_{GS}



Local POD+Galerkin projection, fundamentals.



Question: Snapshots calculated in 0<t<T. Will the POD manifold also describe the dynamics for t>T?

Answer: Yes, provided that some additional POD modes are retained. Retained modes are primary modes.

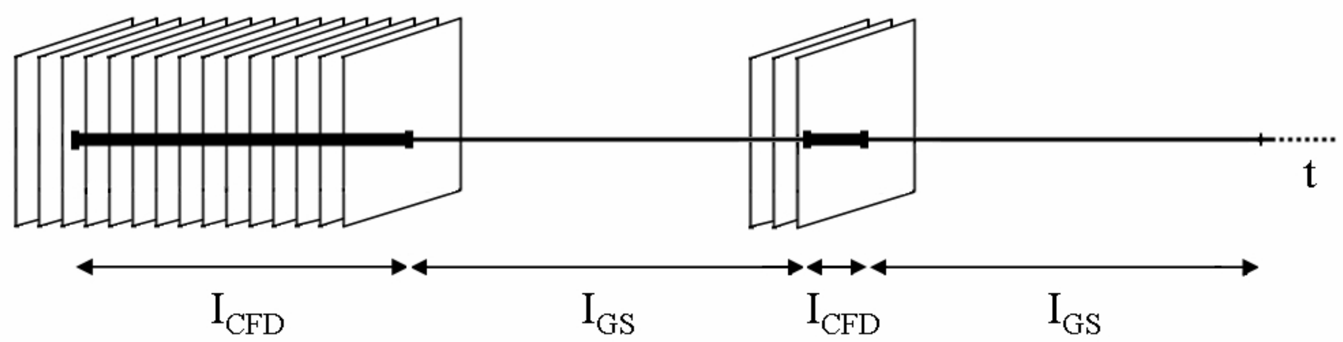
Question: Is it possible to estimate a priori the error of a Galerkin approximation?

Answer: Yes, using some additional higher order modes (secondary modes).

Local POD+Galerkin projection, main strategy.

Strategy:

- CFD integrate the eq'ns in an initial CFD-interval, extract snapshots, and calculate a POD manifold.
- Project the eq'ns on the POD manifold and integrate the resulting Galerkin system in a GS-interval until the approximation fails (a priori error estimate).
- Repeat steps 1 and 2 until the final value of t is reached.



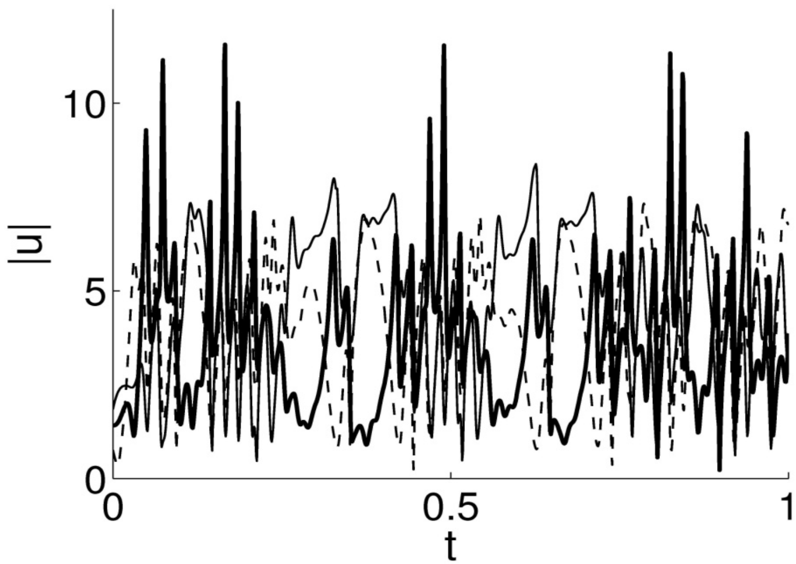
Local POD+Galerkin projection, improvements.

- The POD manifold calculated only at the first CFD-interval (just updated in subsequent CFD intervals). A snapshots library would shorten the first CFD interval.
- The length of the CFD-intervals automatically chosen by numerical trial and error.
- Galerkin projection made using a non-standard inner product based on a few mesh points.
- A second Galerkin system with a larger # of POD modes also integrated. Comparison of the solution with both Galerkin systems provides a second a priori error test.
- Higher order modes instability automatically avoided.
- Galerkin projection of either the exact governing eq'ns or the CFD code.

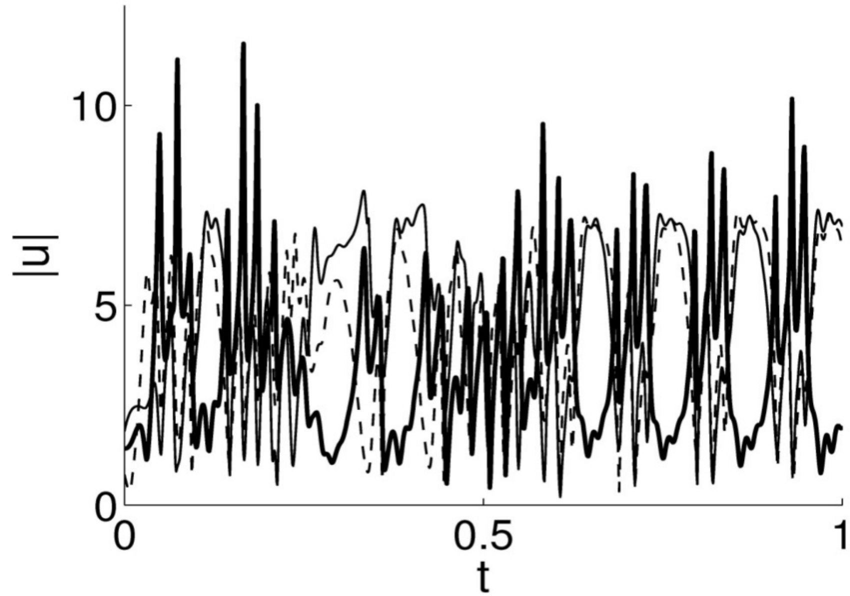
Local POD+Galerkin projection, example #1

$$u_t = (1+i\alpha)u_{xx} + \mu u - (1+i\beta)|u|^2 u$$

Complex Ginzburg-Landau eq'n $(\alpha = -2, \mu = 90, \beta = 14)$ in transient chaos.
|u| at x=1/4 (thin, solid), x=1/2 (thick), and x=3/4 (thin, dashed):



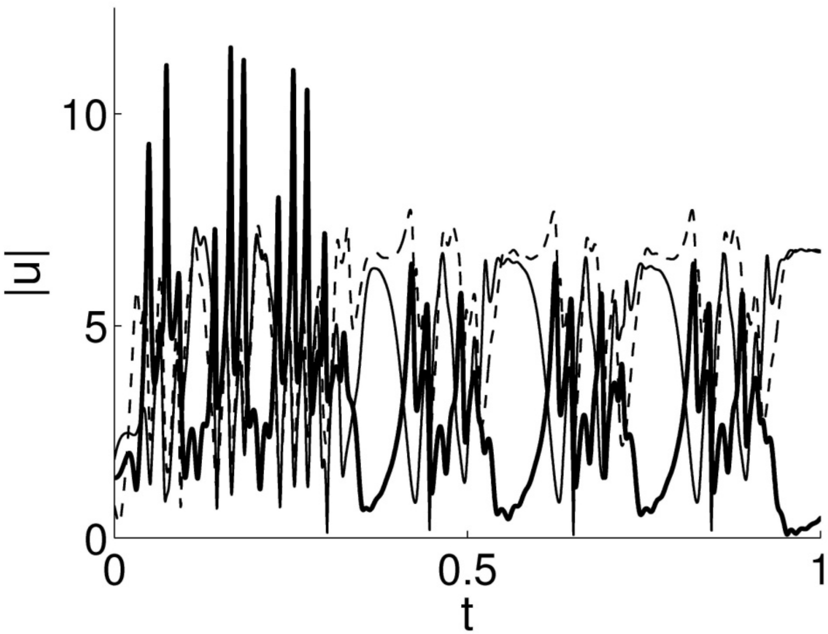
1,000 mesh points

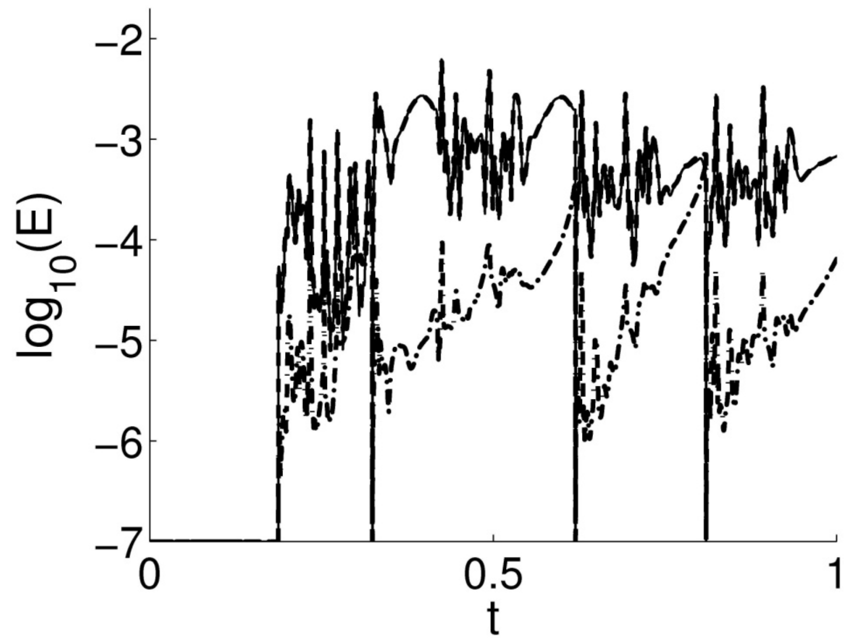


2,000 mesh points

Local POD+Galerkin projection, example #1 (cont'ed)

ROM for GLE in transient chaos.

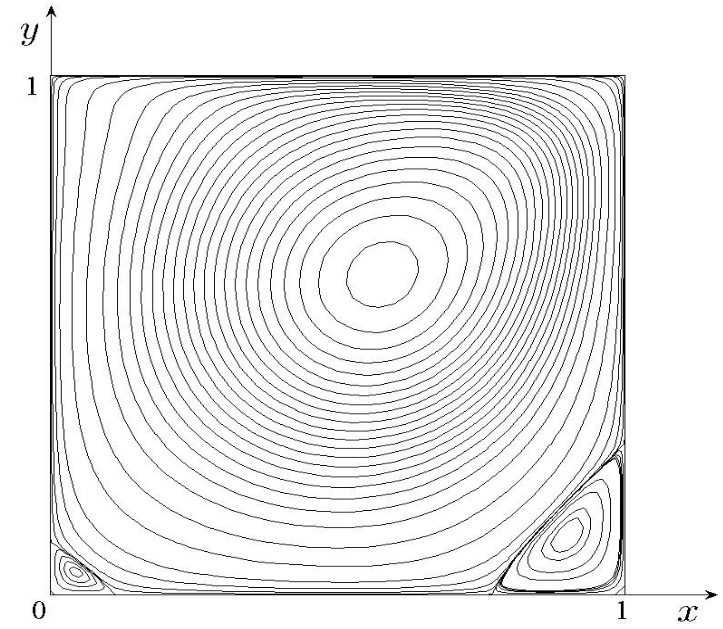




Left: |u| at x=1/4 (thin, solid), 1/2 (thick, solid), and 3/4 (thin, dashed).
Right: Estimated (thin, solid) and exact (thick, dashed) RMS errors using 29 modes: indistinghisable;
RMS error using 38 modes (dash-dotted).
Theoretical compression: 3.4.

Local POD+Galerkin projection, example #2

Standard lid cavity incompressible fluid dynamics in a 2D box whose upper wall is steadily moving (2D NS eq'ns+non-homogeneous BCs). At moderate Reynolds #, the flow relaxes to a steady state. Hopf bifurcation at Re= 7,972. Additional bifurcations at larger Re.



Pulsating lid cavity (upper wall moving back and forth): a paradigm to test CFD. Flow "complexity" at moderate Re.

The pulsating cavity is quite demanding: (i) non-steadiness affects the boundary layer near the moving wall, which increases the number of POD modes; (ii) the bulk velocity is much smaller than forcing velocity, which requires stronger precision to maintain relative errors.

Local POD+Galerkin projection, example #2 (cont'ed)

Pulsating cavity:

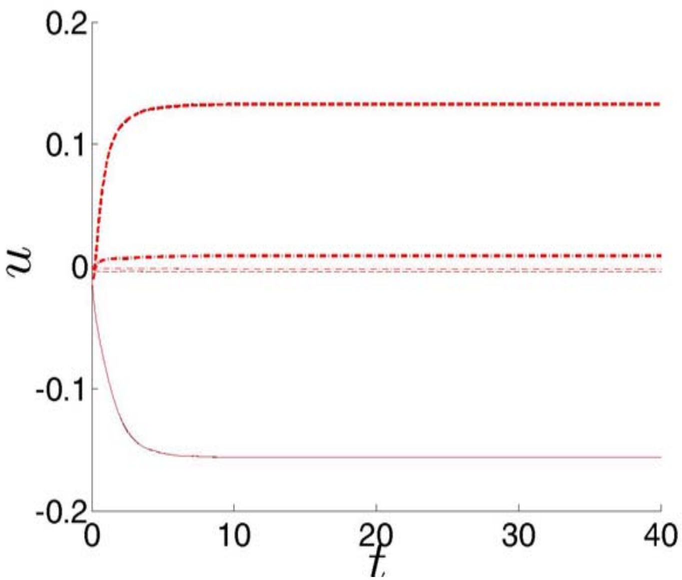
- CFD using a rough/fast numerical code based on finite differences.
- Three grids used for the two velocity components and pressure; rough switch between them.
- Upper corner singularity smoothed out to facilitate comparison with spectral CFD.
- Fourth order terms added to viscous terms to allow for 1D-factorization of the Laplacian.
- Poisson eq'n for pressure integrated with unphysical BCs.
- Lid upper wall BCs poorly satisfied.
- Standard projection of the NS eq'ns does not produce good results in local POD+Galerkin projection.
- Instead, a projection is made that accounts for the way in which discretization is made in CFD.

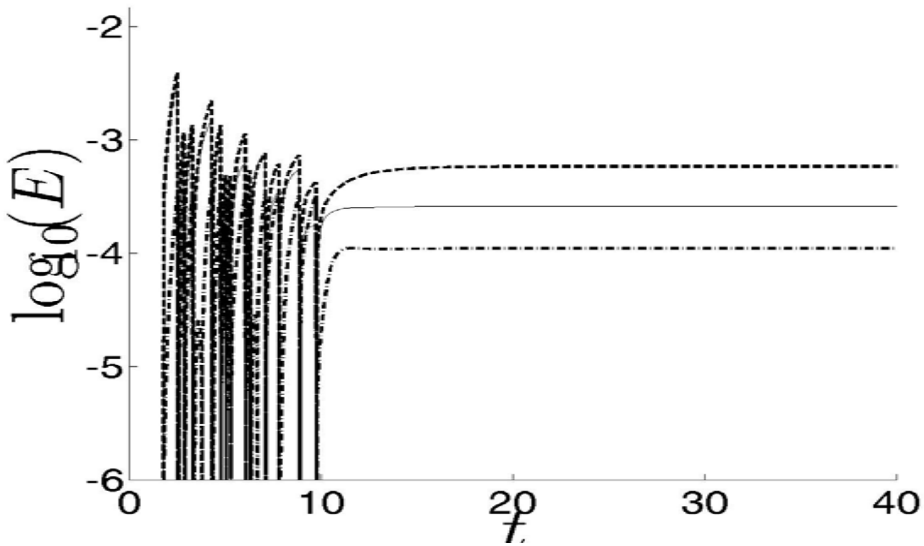
AEROMATH
E.T.S.I. Aeronáuticos

POLITÉCNICA

Local POD+Galerkin projection, example #2

Steady motion of the upper wall. Re=100.

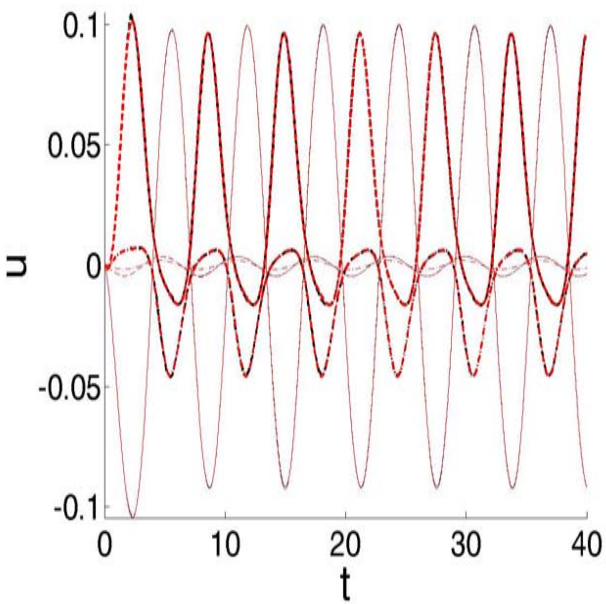


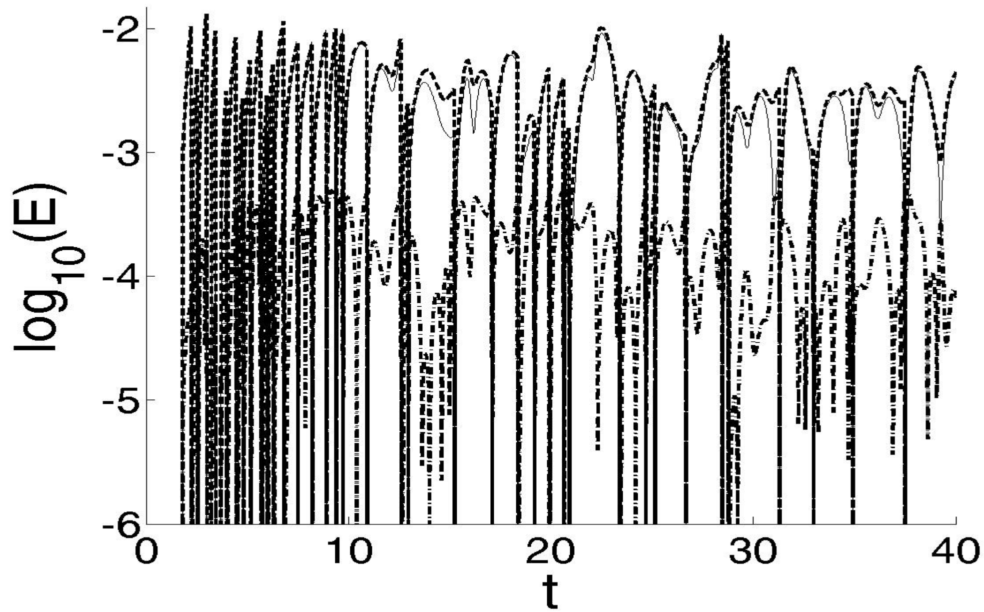


Left: CFD/ROM horizontal velocity at five points (upper/lower corners, center).
Right: log(E) with 5 modes, estimated (thin, solid) and exact (thick, dashed); log(E) with 12 modes (dash-dotted).
Theoretical compression: 17; CPU compression: 6.1

Local POD+Galerkin projection, example #2 (cont'ed)

Periodic motion of the upper wall. Re=100.

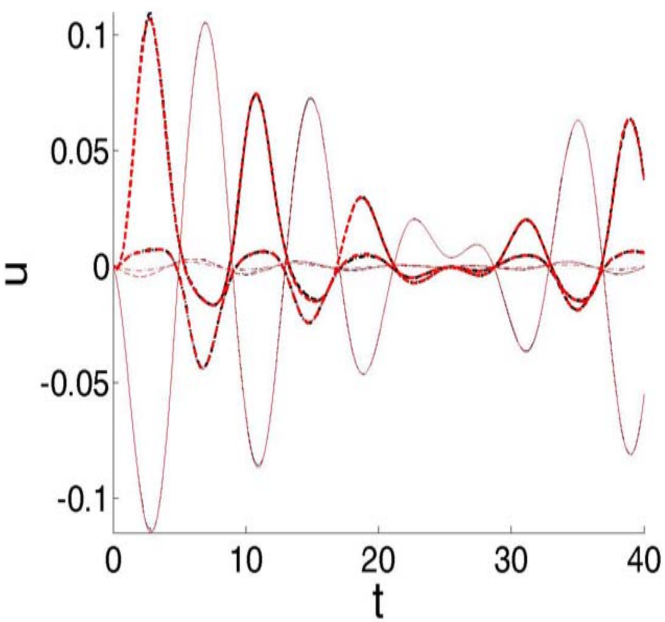


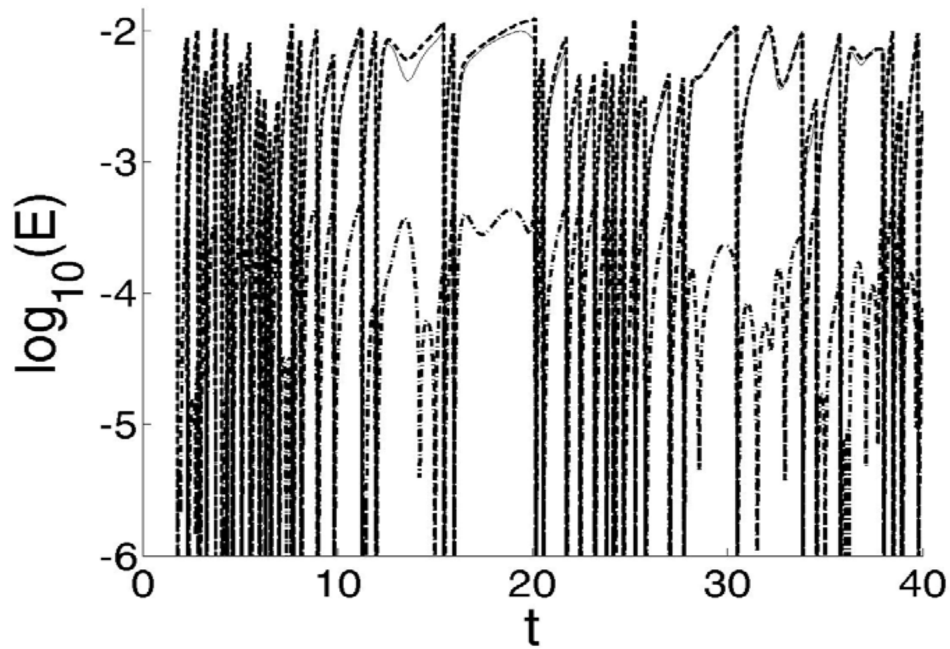


Left: CFD/ROM horizontal velocity at five points (upper/lower corners, center).
Right: log(E) with 7 modes, estimated (thin, solid) and exact (thick, dashed);
Log(E) with 11 modes (dash-dotted).
Theoretical ompression: 10.1; CPU compression: 8.1

Local POD+Galerkin projection, example #2 (cont'ed)

Quasi-periodic motion of the upper wall. Re=100.





Left: CFD/ROM horizontal velocity at five points (upper/lower corners, center).
Right: log(E) with 9 modes, estimated (thin, solid) and exact (thick, dashed);
log(E) with 18 modes (dash-dotted).
Theoretical compression: 8.2; CPU compression: 6.4

Summarizing:

- Efficient (cheap, robust, and precise) ROMs can be constructed using local POD+Galerkin projection +some basic, reasonable improvements based on simple ideas.
- Large theoretical/CPU compression, even using crude software (improvement in progress). Amenable to parallelization and specific hardware.
- Resulting errors comparable to CFD errors (checked using a spectral code).
- Computational effort mainly due to the first calculation of snapshots. Improved using snapshots libraries (in progress).
- Both precise (projecting NS eq'ns) and rough (accounting for the CFD discretization) solvers can be dealt with.
- Non-zero BCs appropriately accounted for (without any change of variable). More flexible projection needed to directly account for general BCs (in progress).